



```
/*-----  
* as_busyout.c - Slot busyout command processing  
*  
* Nov 1999 Padmaja Gogineni  
*  
* Copyright (c) 1999, 2000, 2002 by cisco Systems, Inc.  
* All rights reserved.  
*-----  
*/
```

```
#include "master.h"  
#include "interface.h"  
#include <string.h>  
#include COMP_INC(cisco, ciscolib.h)  
#include COMP_INC(kernel, clock.h)  
#include "interface_private.h"  
#include "../if/if_controller.h"  
#include "../les/if_les.h"  
#include "../nas/nas_common.h"  
#include "../src-7k-as5400/plugin_dfc.h"  
#include "../src-7k-as5400/slot_dfc.h"  
#include "../src-7k-as5400/as5400_dfc.h"  
#include "../nas/parser_defs_busyout.h"  
#include "../nas/busyout_chain.h"  
#include "../les/csm_registry.h"  
#include "../les/isdn_bri.h"  
#include "../if/dsxl.h"  
#include "../isdn/isdn_registry.h"  
#include "../wan/isdn.h"  
#include "../if/dsxl_cas.h"  
#include "as_utils.h"  
#include "if_as_dsxl.h"
```

```
/* includes for parsing */  
#include "config.h"  
#include "parser.h"
```

```
/*  
* Slot busy out command()  
*  
* Parameters:  
*  
* csb - parser information block  
*  
* Returns:  
*  
* void  
*  
* Description: validate the command and call slot specific busyout plugin  
*/
```

```
void slot_busyout_command (parseinfo *csb)  
{  
    plugin_dfc_t *plugin_dfc;  
    ushort slot_no;  
    boolean busyout;  
  
    slot_no = GETOBJ(int, 2);
```

RECEIVED
SEP 11 2003
Technology Center 2100

```

if (!feature_card_exists(slot_no)) {
    if (carrier_card_exists(slot_no)
        && dfc_bit_bucket_on(slot_no)) {
        printf("OIR cleanup in progress for slot %d..., "
            "command ignored, try later!\n", slot_no);
    } else {
        printf("No dfc present in slot %d\n", slot_no);
    }
    return;
}

plugin_dfc = get_plugin_dfc(slot_no);
if (plugin_dfc == NULL) {
    /*
     * Should never happen
     */
    BASSERT(FALSE);
    return;
}

busyout = csb->sense;

if (busyout == FALSE) {
    if (!IS_SLOT_BUSYOUT_MODE(slot_no) && dfc_power_good(slot_no)) {
        printf("Slot %d is not currently in busyout mode\n", slot_no);
        return;
    }
} else {
    if (IS_SLOT_BUSYOUT_MODE(slot_no)) {
        printf("Slot %d already in busyout mode\n", slot_no);
        return;
    }
}

if (busyout && !dfc_power_good(slot_no)) {
    printf("DFC in slot %d is powered down.\n", slot_no);
    return;
} else {
    MARK_SLOT_BUSYOUT_IN_PROGRESS(slot_no, busyout);
    printf("%s in progress for %d", (busyout ? "Busyout" : "Unbusyout"),
        slot_no);
}

/*
 * Update ok/maintenance LED to "yellow".
 */
DFC_LED_UPDATE(slot_no);

if (busyout == FALSE) {
    MARK_SLOT_BUSYIED_OUT(slot_no, FALSE);
}

switch (csb->which) {
case BUSYOUT_SERVICE:
    plugin_dfc->slot->exec_slot_busyout = TRUE;

```

```

        plugin_dfc->plugin_ft->plugin_busyout(plugin_dfc, busyout);
        break;
    default:
        BASSERT(FALSE);
        break;
    }
}

/*
 * show_busyout_command()
 *
 * Parameters:
 *
 * csb - parser information block
 *
 * Returns:
 *
 * void
 *
 * Description: Invoke slot specific show busyout plugin
 */
void show_busyout_command (parseinfo *csb)
{
    int busyout_slot, dfc_slot;
    plugin_dfc_t *plugin_dfc;

    busyout_slot = GETOBJ(int, 2);

    /*
     * Invoke slot dependent busyout status routines
     */
    if (busyout_slot == -1) {
        FOR_ALL_AS_SLOTS(dfc_slot) {
            if (feature_card_operational(dfc_slot)) {
                plugin_dfc = get_plugin_dfc(dfc_slot);
                if (plugin_dfc)
                    plugin_dfc->plugin_ft->plugin_show_busyout(plugin_dfc);
            }
        }
    } else {
        if (!feature_card_exists(busyout_slot)) {
            if (carrier_card_exists(busyout_slot)
                && dfc_bit_bucket_on(busyout_slot)) {
                printf("OIR cleanup in progress for slot %d.\n",
                    busyout_slot);
            } else {
                printf("No DFC present in slot %d.\n", busyout_slot);
            }
        }
        return;
    }

    if (!dfc_power_good(busyout_slot)) {
        printf("DFC in slot %d is powered down.\n", busyout_slot);
        return;
    }
}

```

```
}  
  
plugin_dfc = get_plugin_dfc(busyout_slot);  
if (plugin_dfc) {  
    plugin_dfc->plugin_ft->plugin_show_busyout(plugin_dfc);  
}  
}
```